MOODLE STAGING SITE IMPLEMENTATION DOCUMENTATION

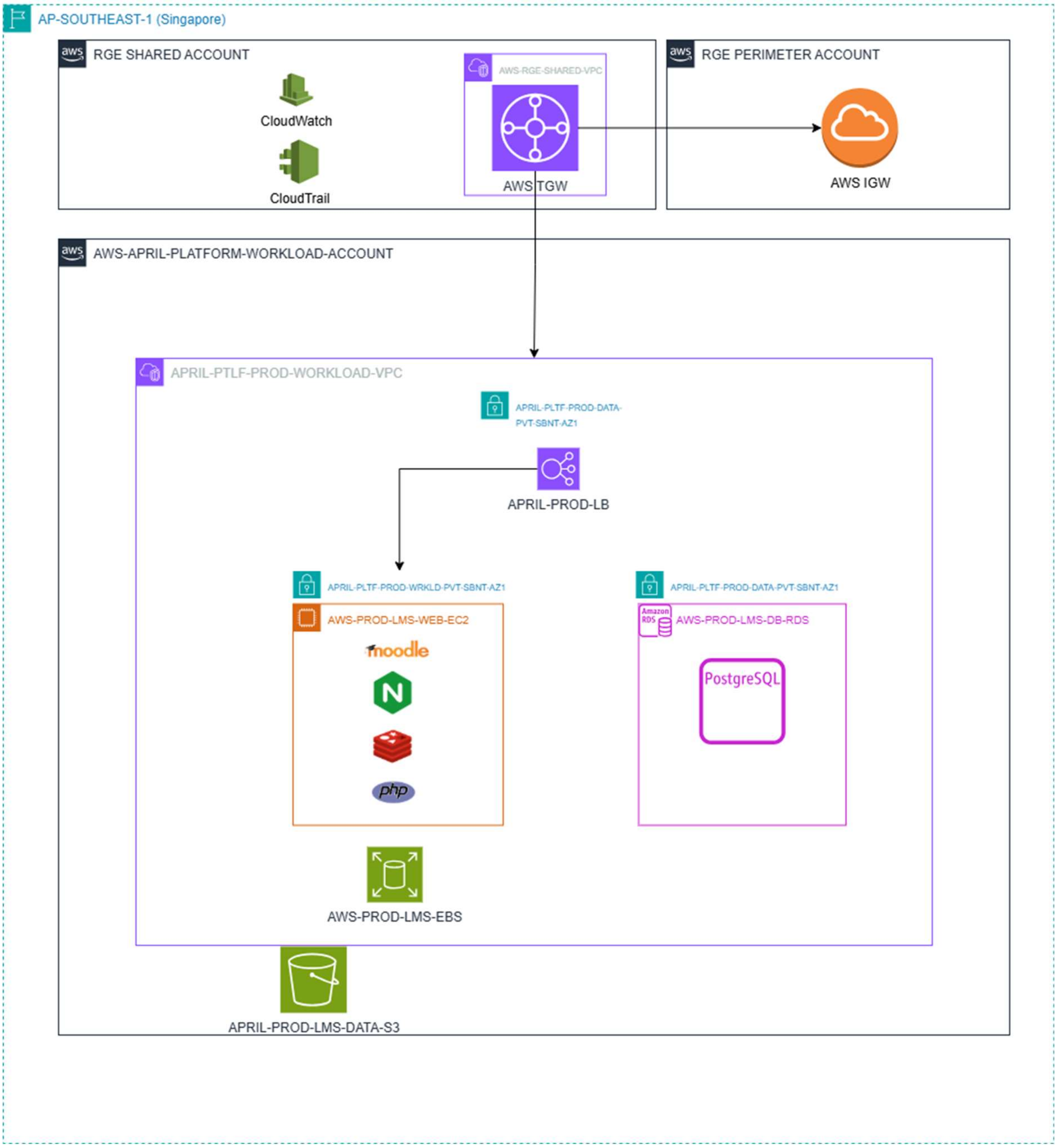
PT. Prima Transportasi Service Indonesia

PT Pasifik Cipta Mandiri
Komplek Puri Niaga III, Jl. Puri Kencana Blok M8 No.3F, RT.3/RW.7, Kembangan Sel., Kec. Kembangan,
Kota Jakarta Barat, Daerah Khusus Ibukota Jakarta 11610

TABLE OF CONTENTS

A. MOODLE STAGING ARCHITECTURE

In implementing the Moodle Staging Site for Prima Transportasi Service Indonesia, we have implemented the architecture according to the provided reference.



B. MOODLE STAGING SYSTEM SPECIFICATIONS

Staging Workload					
Instance	**Specification**		**Number Service**	**Notes**	**Recommendation**
Moodle apps + Moodledata (AWS EC2)	vCPU (Core)	2	1		
	RAM (GB)	4			
	Storage (GB)	256			OS: Amazon Linux 2023 64-bit
Moodle Database (AWS RDS) ***db.t4g.medium***	vCPU (Core)	2	1		
	RAM (GB)	4			
	Storage (GB)	100			
AWS Simple Storage Service (S3)	Type	Standard and Data Transfer	1		

C. MOODLE STAGING SYSTEM INFORMATION

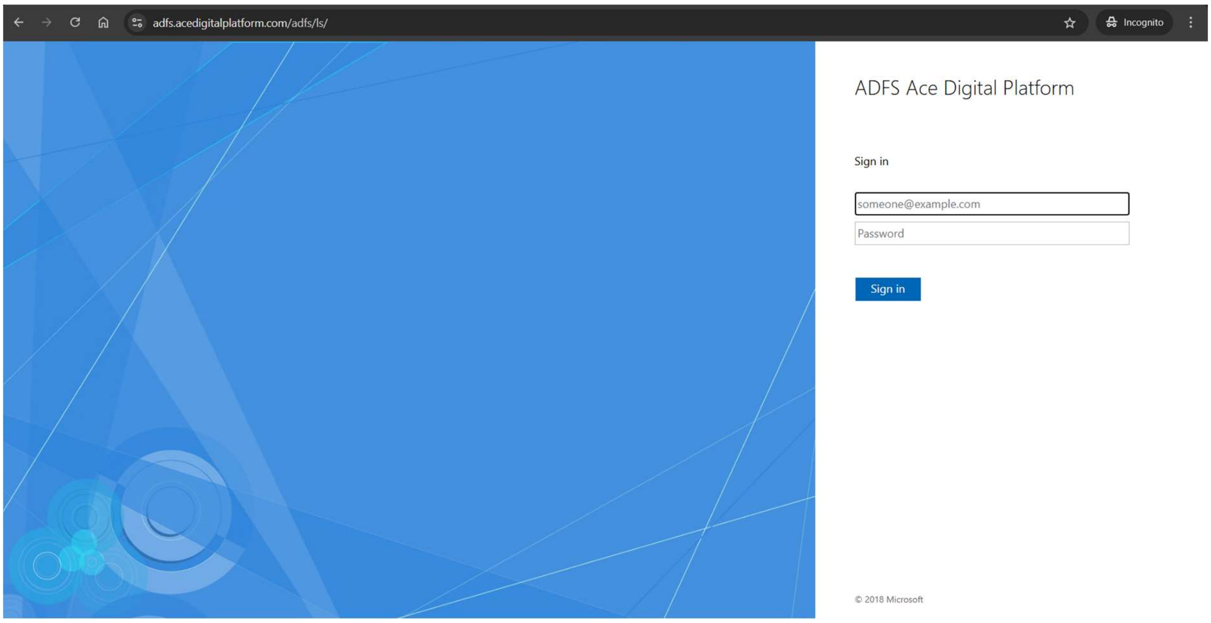
Instance Name	IP Address/Endpoint	Operating System/Engine System
AWS-STAGE-LMS-WEB-EC2	10.101.185.54	Amazon Linux 2023
Aws-stage-lms-db-rds	aws-stage-lms-db-rds.chgy6gyc8kng.ap-southeast-1.rds.amazonaws.com	Postgresql 16.8 R1

D. PRE-INSTALLATION PROCEDURES

Before installing Moodle on the prepared server, several prerequisites must be addressed, including creating EC2 instance, creating RDS instance, installing package dependencies, configuring PHP, configuring the database, and other necessary steps.

1. AWS Portal Access

- Open the AWS console through the URL https://acedigitalplatform.awsapps.com/start/ and enter credentials for the globalnet account



adfs.acedigitalplatform.com/adfs/ls/

ADFS Ace Digital Platform

Sign in

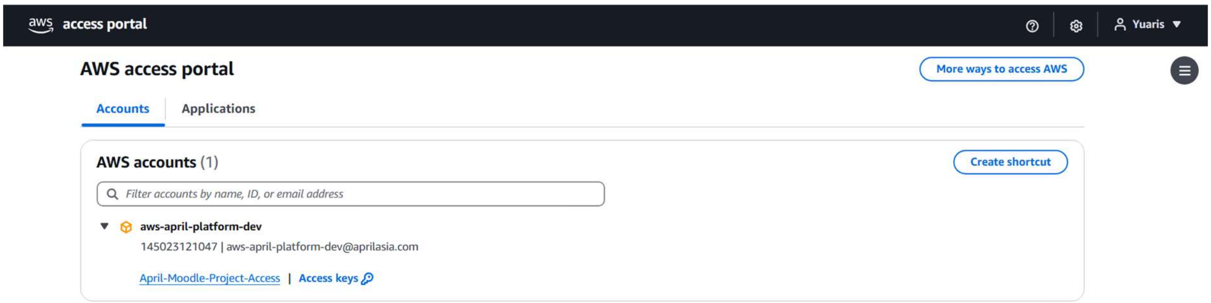
someone@example.com

Password

Sign in

© 2018 Microsoft

- In the AWS Access Portal, select aws-april-platform-dev > April-Moodle-Project-Access



AWS access portal

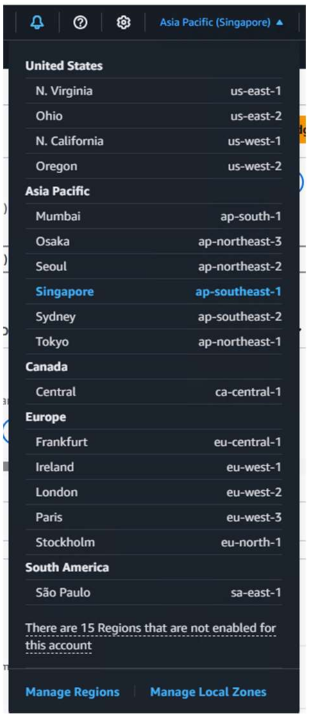
AWS accounts (1)

aws-april-platform-dev

145023121047 | aws-april-platform-dev@aprilasia.com

April-Moodle-Project-Access | Access keys

- Ensure the project is set to the Singapore region



Asia Pacific (Singapore)

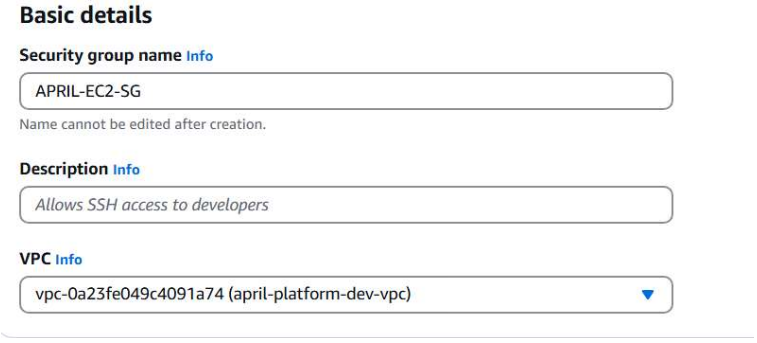
United States	
N. Virginia	us-east-1
Ohio	us-east-2
N. California	us-west-1
Oregon	us-west-2
Asia Pacific	
Mumbai	ap-south-1
Osaka	ap-northeast-3
Seoul	ap-northeast-2
Singapore	ap-southeast-1
Sydney	ap-southeast-2
Tokyo	ap-northeast-1
Canada	
Central	ca-central-1
Europe	
Frankfurt	eu-central-1
Ireland	eu-west-1
London	eu-west-2
Paris	eu-west-3
Stockholm	eu-north-1
South America	
São Paulo	sa-east-1

There are 15 Regions that are not enabled for this account

Manage Regions | Manage Local Zones

2. Creating Security Group

- Go to EC2 > Security Group > Create Security Group
- Fill in the Security Group details



Basic details

Security group name Info

APRIL-EC2-SG

Name cannot be edited after creation.

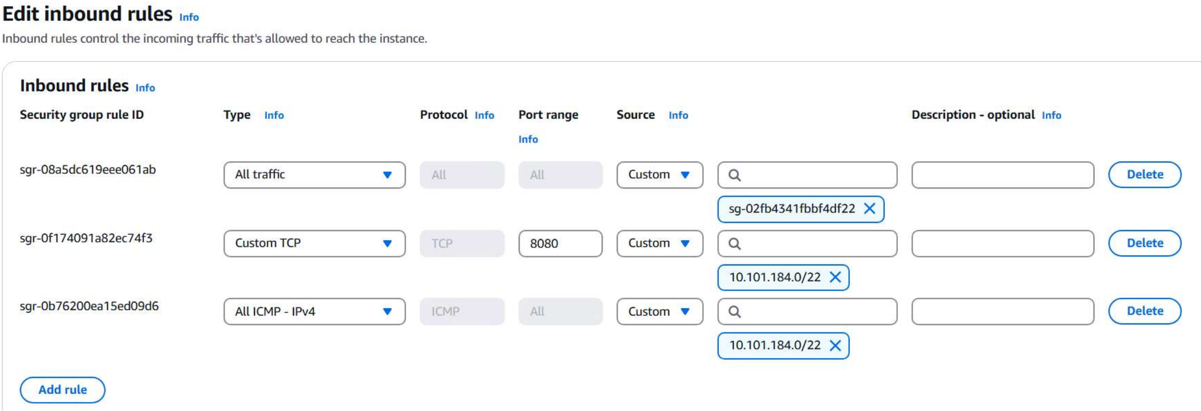
Description Info

Allows SSH access to developers

VPC Info

vpc-0a23fe049c4091a74 (april-platform-dev-vpc)

- Configure Inbound Rules



Edit inbound rules Info

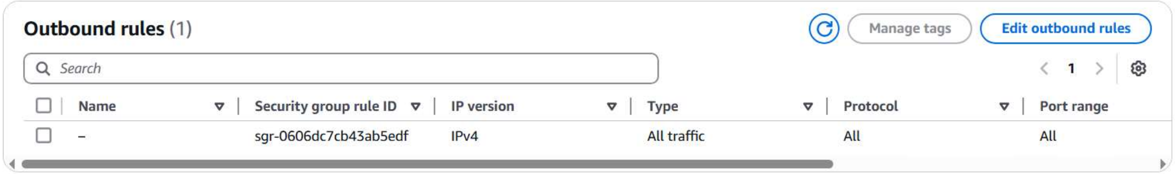
Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional	
sgr-08a5dc619eee061ab	All traffic	All	All	Custom	sg-02fb4341fbbf4df22		Delete
sgr-0f174091a82ec74f3	Custom TCP	TCP	8080	Custom	10.101.184.0/22		Delete
sgr-0b76200ea15ed09d6	All ICMP - IPv4	ICMP	All	Custom	10.101.184.0/22		Delete

Add rule

- Configure Outbound Rules



Outbound rules (1)

□	Name	Security group rule ID	IP version	Type	Protocol	Port range
□	–	sgr-0606dc7cb43ab5edf	IPv4	All traffic	All	All

- Click “Create security group”

3. Creating EC2 Instance

- In the AWS Console, select EC2 > Instances > Launch Instances
- Configure the EC2 according to the specifications determined, for this implementation we used:

➤ *Name and tags*

- Name: APRIL-STAGE-LMS-WEB-EC2
- Tags: Fill according to requirements

Key	Value
rge:businessgroup	april
rge:april:department	lms-team
rge:april:projectname	moodle
rge:april:environment	dev
rge:april:owner	yuaris_arham@globalnet.lcl

➤ *Application and OS Images (AMI)*

- Select **Amazon Linux 2023 AMI**

➤ *Instance type*

- t3.medium

➤ *Network settings*

- VPC & Subnet: adjust to the VPC used by the Security Group
- **Auto-assign public IP**: check **disable**
- **Firewall (security groups)**:
  - Select existing security group
  - Choose the previously created SG (APRIL-EC2-SG)

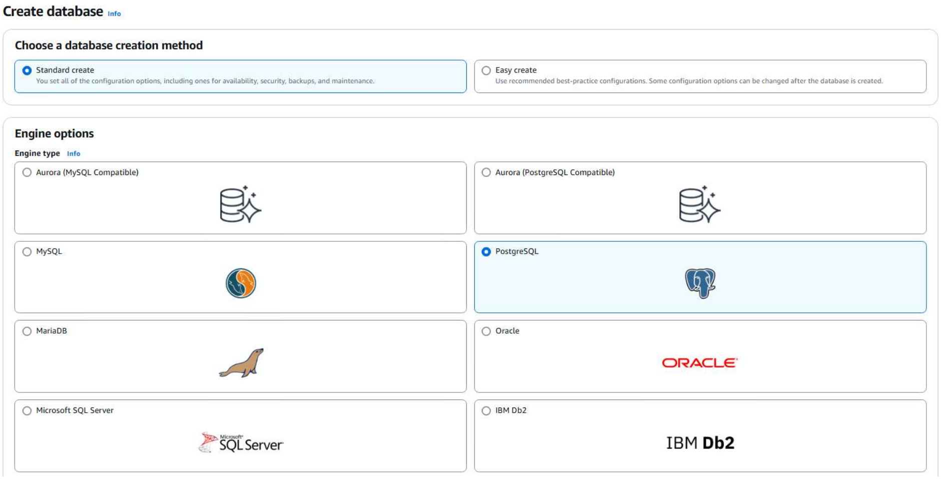
- *Storage (optional)*

  - Set: 256 GB gp3 SSD

4. Creating RDS Instance

- Search for "**RDS**" in the top search bar, click on the result.
- In the left sidebar, click **Databases**.
- Click the **"Create database"** button.
- Fill in and adjust the configuration details according to specifications

Name	Value
Engine Options	PostgreSQL
Engine Version	Postgresql 16.8 R1
Templates	Dev/Test
DB Instance Identifier	april-stage-lms-db-rds
DB Instance Class	db.t4g.medium
Storage	100 GB gp3
VPC	Select the same VPC as the EC2



Create database Info

Choose a database creation method

● Standard create
You set all of the configuration options, including ones for availability, security, backups, and maintenance.

○ Easy create
Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type Info

○ Aurora (MySQL Compatible)

○ Aurora (PostgreSQL Compatible)

○ MySQL

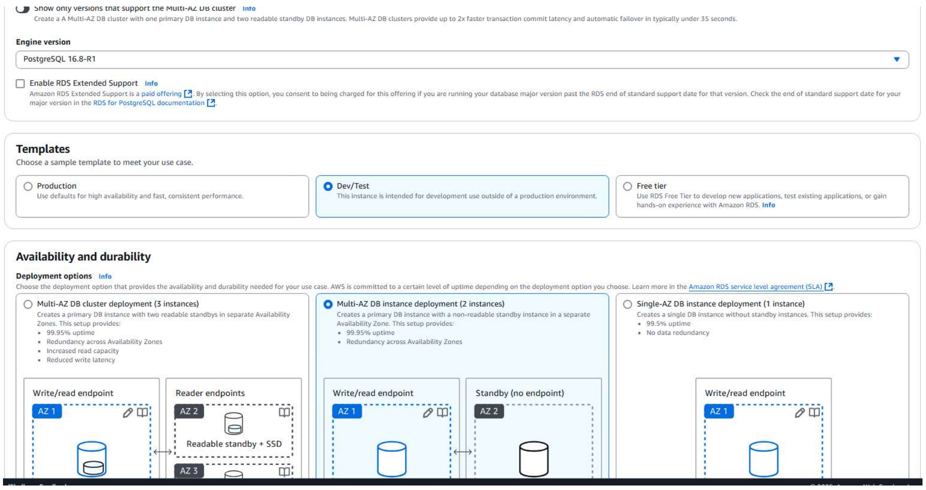
● PostgreSQL

○ MariaDB

○ Oracle

○ Microsoft SQL Server

○ IBM Db2



Show only versions that support the Multi-AZ DB cluster Info
Create a A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Engine version
PostgreSQL 16.8-R1

□ Enable RDS Extended Support Info
Amazon RDS Extended Support is a paid offering. By selecting this option, you consent to being charged for this offering if you are running your database major version past the RDS end of standard support date for that version. Check the end of standard support date for your major version in the RDS for PostgreSQL documentation.

Templates
Choose a sample template to meet your use case.

○ Production
Use defaults for high availability and fast, consistent performance.

● Dev/Test
This instance is intended for development use outside of a production environment.

○ Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS. Info

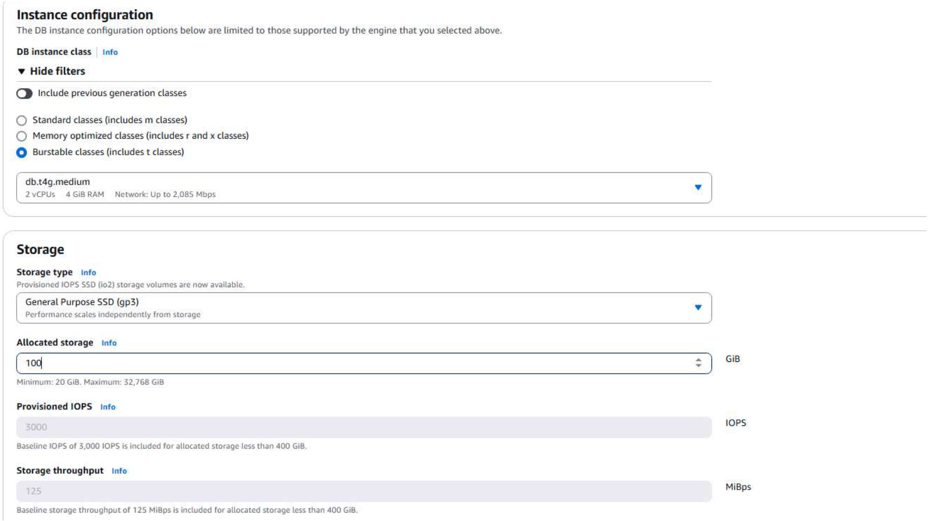
Availability and durability

Deployment options Info
Choose the deployment option that provides the availability and durability needed for your use case. AWS is committed to a certain level of uptime depending on the deployment option you choose. Learn more in the Amazon RDS service level agreement (SLA).

○ Multi-AZ DB cluster deployment (3 instances)
Creates a primary DB instance with two readable standbys in separate Availability Zones. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones
- Increased read capacity
- Reduced write latency

● Multi-AZ DB instance deployment (2 instances)
Creates a primary DB instance with a non-readable standby instance in a separate Availability Zone. This setup provides:
- 99.95% uptime
- Redundancy across Availability Zones

○ Single-AZ DB instance deployment (1 instance)
Creates a single DB instance without standby instances. This setup provides:
- 99.5% uptime
- No data redundancy



Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

DB instance class | Info

▼ **Hide filters**

Include previous generation classes

○ Standard classes (includes m classes)

○ Memory optimized classes (includes r and x classes)

● Burstable classes (includes t classes)

db.t4g.medium
2 vCPUs 4 GiB RAM Network: Up to 2,085 Mbps

Storage

Storage type Info

Provisioned IOPS SSD (io2) storage volumes are now available.

General Purpose SSD (gp3)
Performance scales independently from storage

Allocated storage Info

100 GiB

Minimum: 20 GiB. Maximum: 32,768 GiB

Provisioned IOPS Info

3000 IOPS

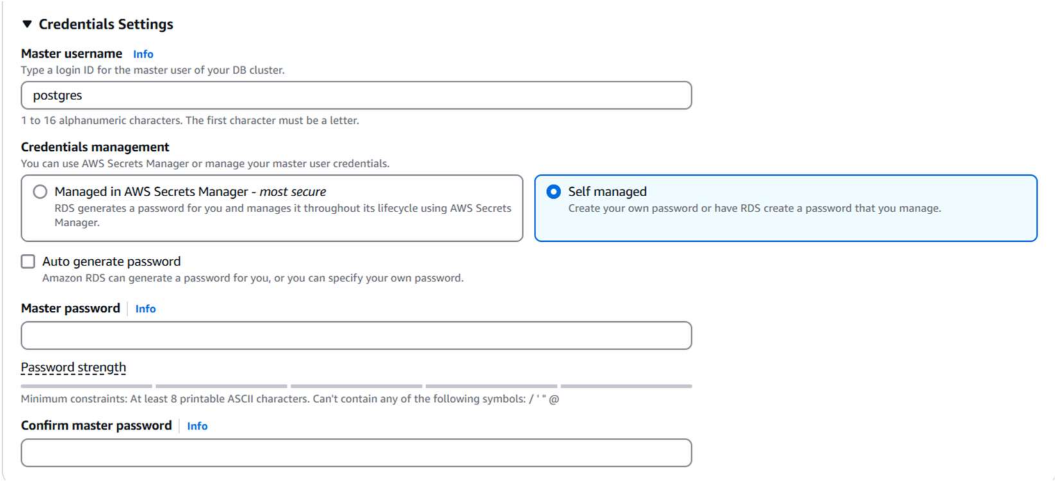
Baseline IOPS of 3,000 IOPS is included for allocated storage less than 400 GiB.

Storage throughput Info

125 MiBps

Baseline storage throughput of 125 MiBps is included for allocated storage less than 400 GiB.

- Fill in the database credentials such as master username and master password as needed



▼ **Credentials Settings**

Master username Info

Type a login ID for the master user of your DB cluster.

postgres

1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management

You can use AWS Secrets Manager or manage your master user credentials.

○ Managed in AWS Secrets Manager - *most secure*
RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.

● Self managed
Create your own password or have RDS create a password that you manage.

□ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

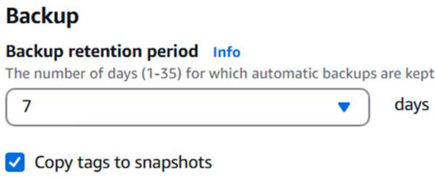
Master password | Info

Password strength

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password | Info

- Additional Configuration
  - Set Backup Retention



Backup

Backup retention period Info

The number of days (1-35) for which automatic backups are kept.

7 days

☑ Copy tags to snapshots

- For the tags section, use the same tags as the EC2 Instance
- Click "Create Database".

5. Creating S3 Bucket

- In the AWS Console, open S3 Service > Create Bucket
- For S3 Bucket configuration:

➤ General Configuration

- Bucket name: April-stage-lms-data-s3

➤ *Object Ownership*

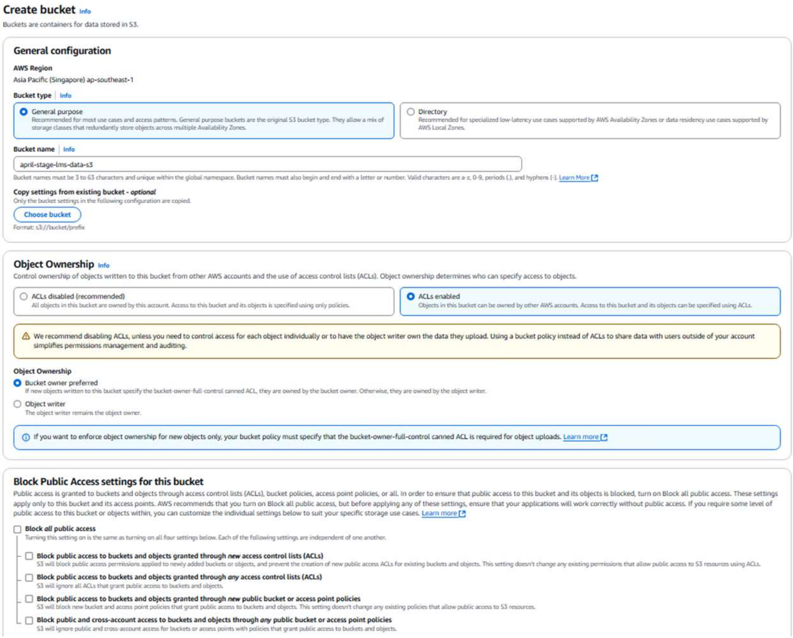
- ACLs-enabled
- Object Owner Prefered

➤ *Block Public Access settings for this bucket*

- Uncheck Block all public access

➤ *Tags*

- Use the same tags as other resources according to the standard
- The configuration used can be modified at any time in the staging environment to adjust to applicable policies.



Create bucket Info

Buckets are containers for data stored in S3.

General configuration

AWS Region

Asia Pacific (Singapore) ap-southeast-1

Bucket type | Info

● General purpose
Recommended for most use cases and access patterns. General purpose buckets are the original S3 bucket type. They allow a mix of storage classes that redundantly store objects across multiple Availability Zones.

○ Directory
Recommended for specialized low-latency use cases supported by AWS Availability Zones or data residency use cases supported by AWS Local Zones.

Bucket name | Info

april-stage-lms-data-s3

Bucket names must be 3 to 63 characters and unique within the global namespace. Bucket names must also begin and end with a letter or number. Valid characters are a-z, 0-9, periods (.), and hyphens (-). Learn More

Copy settings from existing bucket - *optional*

Only the bucket settings in the following configuration are copied.

Format: s3://bucket/prefix

Object Ownership Info

Control ownership of objects written to this bucket from other AWS accounts and the use of access control lists (ACLs). Object ownership determines who can specify access to objects.

○ ACLs disabled (recommended)
All objects in this bucket are owned by this account. Access to this bucket and its objects is specified using only policies.

● ACLs enabled
Objects in this bucket can be owned by other AWS accounts. Access to this bucket and its objects can be specified using ACLs.

⚠ We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.

Object Ownership

● Bucket owner preferred
If new objects written to this bucket specify the bucket-owner-full-control canned ACL, they are owned by the bucket owner. Otherwise, they are owned by the object writer.

○ Object writer
The object writer remains the object owner.

ⓘ If you want to enforce object ownership for new objects only, your bucket policy must specify that the bucket-owner-full-control canned ACL is required for object uploads. Learn more

Block Public Access settings for this bucket

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. Learn more

□ **Block *all* public access**
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- □ **Block public access to buckets and objects granted through *new* access control lists (ACLs)**
  S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- □ **Block public access to buckets and objects granted through *any* access control lists (ACLs)**
  S3 will ignore all ACLs that grant public access to buckets and objects.
- □ **Block public access to buckets and objects granted through *new* public bucket or access point policies**
  S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- □ **Block public and cross-account access to buckets and objects through *any* public bucket or access point policies**
  S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

6. Installing Package Dependencies

- Connect to EC2 Instance
- Install NGINX and PHP on the Moodle server. Run the following command using the root user.

```
sudo dnf install -y nginx php php-fpm php-cli php-common php-pgsql
php-curl php-zip php-gd php-intl php-soap php-mbstring php-xml php-
opcache postgresql16 git php-pear php-ldap
```

- Start and enable NGINX and PHP-FPM

```
sudo systemctl start nginx
sudo systemctl enable nginx
sudo systemctl start php-fpm
sudo systemctl enable php-fpm
```

7. PHP Configuration

- Open the ***/etc /php.ini*** file

```
vi /etc/php.ini
```

- Find the **max_input_vars** syntax in the php.ini file. By default, this configuration has a value of 1000

```
;max_input_vars = 1000
```

- Uncomment this syntax and change the value from 1000 to 5000.

```
max_input_vars = 5000
```

- Restart the php-fpm and NGINX services

```
systemctl restart nginx && systemctl restart php-fpm
```

8. Database Configuration (PostgreSQL)

- Run the following command on the EC2 to access the RDS Instance

```
# psql -h aws-stage-lms-db-rds.chgy6gyc8kng.ap-southeast-
1.rds.amazonaws.com  -U postgres -d postgres -p 5432
```

Note:
-h: RDS instance endpoint
-U: master username set during instance creation
-d: database to access
-p: port used by the database

- Enter the master password after the following line appears:

```
# psql -h aws-stage-lms-db-rds.chgy6gyc8kng.ap-southeast-
1.rds.amazonaws.com  -U postgres -d postgres -p 5432

Password for user postgres:
```

- Create a new user for the Moodle service

```
postgres=# CREATE USER moodleuser WITH PASSWORD 'yourpassword';
```

- Create a new database for Moodle with the previously created user as the owner.

```
postgres=# CREATE DATABASE moodle WITH OWNER moodleuser;
```

9. Downloading Moodle 4.5 Package

- Navigate to the **/var/www/html/** directory.

```
# cd /var/www/html
```

- Download the Moodle 4.5 package from the official Moodle GitHub repository.

```
# git clone -b MOODLE_405_STABLE git://git.moodle.org/moodle.git
```

- Ensure the Moodle package download process has no errors and runs normally. Enter the Moodle directory.

```
# cd moodle
```

- Verify the version of Moodle that was successfully downloaded by opening the ***version.php*** file in the moodle directory.

```
# cat version.php
```

- Make sure the downloaded version is correct

```
$release  = '4.5.4+ (Build: 20250417)';    // Human-friendly
version name
```

10. NGINX Configuration

- Ensure the NGINX service on the EC2 is running

```
# systemctl status nginx
```

- Create a new file at ***/etc/nginx/conf.d/moodle.conf***.

```
# vi /etc/nginx/conf.d/moodle.conf
```

- Adjust the contents of the file as follows.

```
server {
    listen 8080; # Listen on 8080 for ALB health checks and traffic
    #server_name 10.101.185.54;
    server_name lms-dev.fiber.biz.id;
    root /var/www/html/moodle/;
    index index.php index.html index.htm;

    # Add this to properly handle ALB headers
    set_real_ip_from 10.101.184.0/22; # VPC CIDR
    real_ip_header X-Forwarded-For;

    # Add this for ALB health checks
    location /health.php {
        access_log off;
        return 200 "OK\n";
        add_header Content-Type text/plain;
    }

    location / {
       # Add these headers for SSL behind ALB
        proxy_set_header X-Forwarded-Proto $http_x_forwarded_proto;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_set_header Host $http_host;

        try_files $uri $uri/ /r.php;
        #try_files $uri $uri/ /index.php?$query_string;

    }
    location ~ [^/]\.php(/|$) {
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_pass unix:/run/php-fpm/www.sock;
        fastcgi_index index.php;
        include fastcgi_params;

        # Add these for proper SSL handling behind ALB
        fastcgi_param HTTPS on;
        fastcgi_param HTTP_X_FORWARDED_PROTO https;

        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        fastcgi_param PATH_INFO $fastcgi_path_info;

        # Add these for better performance
        fastcgi_buffer_size 128k;
        fastcgi_buffers 4 256k;
        fastcgi_busy_buffers_size 256k;
        fastcgi_read_timeout 300;
    }
    client_max_body_size 500M;
}
```

- Save and exit the file after configuration changes are complete. Restart the NGINX service.

```
# systemctl restart nginx
```

11. Redis Server Configuration

- According to PTSI policy requiring the use of the latest Redis version (Redis 7 at the time of writing this document), while the Amazon Linux 2023 repository only provides Redis version 6, Redis installation needs to be done manually using source code.

- Install system development package

```
# sudo dnf install -y systemd-devel
```

- Download the latest Redis source code

```
# wget https://download.redis.io/redis-stable.tar.gz
```

- Extract the file and enter the directory

```
# tar -xzvf redis-stable.tar.gz
# cd redis-stable
```

- Compile Redis with the system

```
# make USE_SYSTEMD=yes
```

- Install the compiled Redis

```
# make install
```

- Create a systemd service file to manage Redis

```
# vi /etc/systemd/system/redis.service
```

- Write the file with the following configuration

```
[Unit]
Description=Redis In-Memory Data Store
After=network.target

[Service]
Type=notify
User=redis
Group=redis
ExecStart=/usr/local/bin/redis-server /etc/redis/redis.conf
ExecStop=/usr/local/bin/redis-cli shutdown
```

```
Restart=on-failure

[Install]
WantedBy=multi-user.target
```

- Reload systemd and start Redis

```
# sudo systemctl daemon-reload
# sudo systemctl start redis
# sudo systemctl status redis
```

12. Load Balancer Setup

- Connect to the EC2 instance and create a new file for Load Balancer health checks

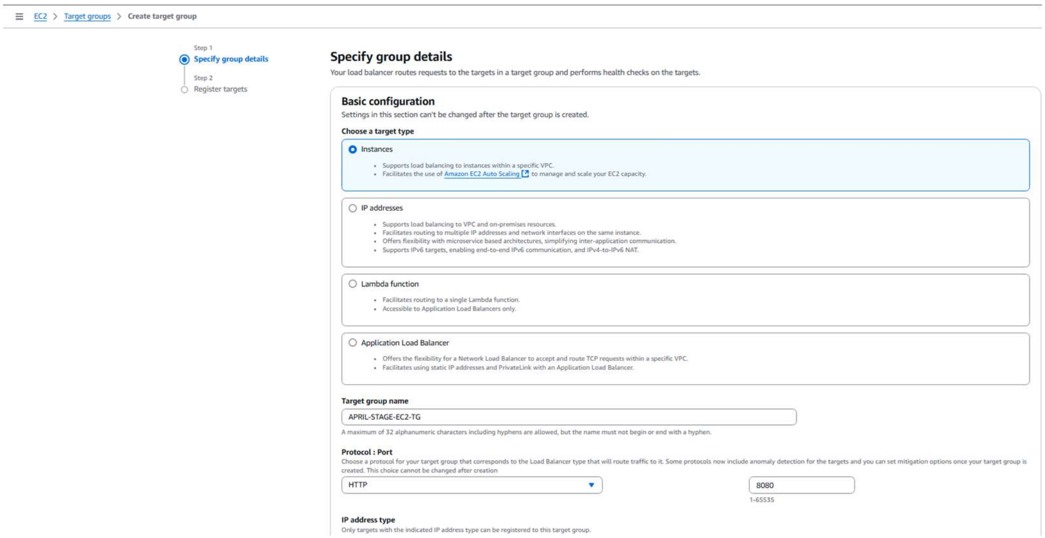
```
# vi /var/www/html/moodle/health.php
```

- Write the file with the following configuration.

```
<?php

header('Content-Type: text/plain');

echo 'OK';
```

- Open EC2 services in AWS
- In the left sidebar, scroll down to the **"Load Balancing"** → **"Target Groups"**
- Click **"Create Target Group"**
- Fill in the details for the target group with the following specifications:

Name	**Value**
Target Type	Instances
Target Group Name	APRIL-STAGE-EC2-TG
Port	HTTP:8080
IP Address Type	IPv4
Protocol	HTTP1
Health Checks	Path: /
VPC	Select the same VPC as the EC2



EC2 > Target groups > Create target group

Step 1
● Specify group details

Step 2
○ Register targets

Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targets.

Basic configuration

Settings in this section can't be changed after the target group is created.

Choose a target type

● Instances
- Supports load balancing to instances within a specific VPC.
- Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.

○ IP addresses
- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.
- Supports IPv6 targets, enabling end-to-end IPv6 communication, and IPv4-to-IPv6 NAT.

○ Lambda function
- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

○ Application Load Balancer
- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

APRIL-STAGE-EC2-TG

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol : Port

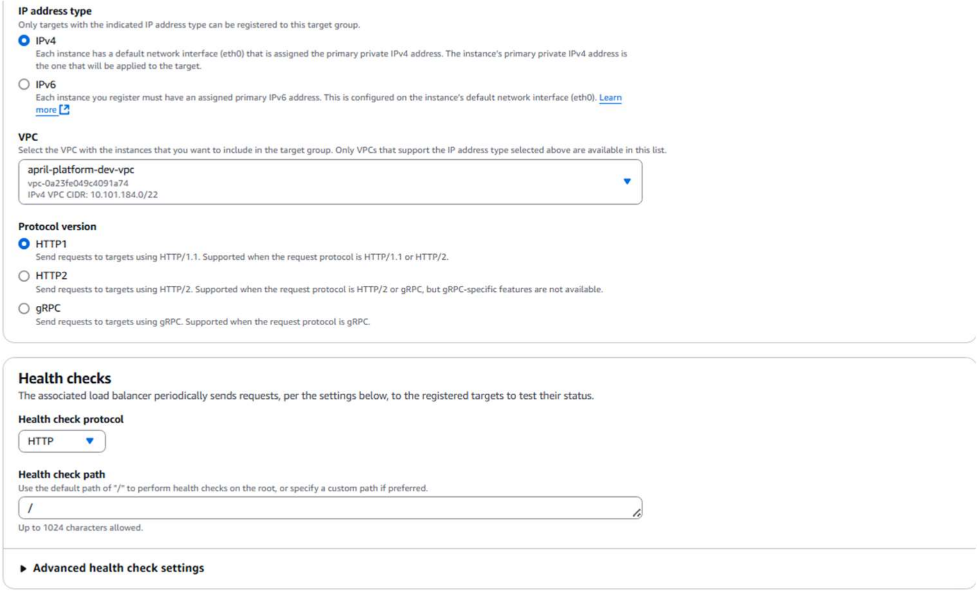
Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation

HTTP ▼ 8080

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.



IP address type

Only targets with the indicated IP address type can be registered to this target group.

● IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

○ IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). Learn more

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

april-platform-dev-vpc
vpc-0a23fe049c4091a74
IPv4 VPC CIDR: 10.101.184.0/22 ▼

Protocol version

● HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

○ HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

○ gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP ▼

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/

Up to 1024 characters allowed.

▸ Advanced health check settings

- Return to the EC2 page
- In the left sidebar, scroll down to the **"Load Balancing"** section → click **"Load Balancer"**
- Click **"Create Load Balancer" > "Network Load Balancer"**
- Fill in the Load Balancer configuration according to the specified specifications, for this implementation we used:

➤ Basic Configuration

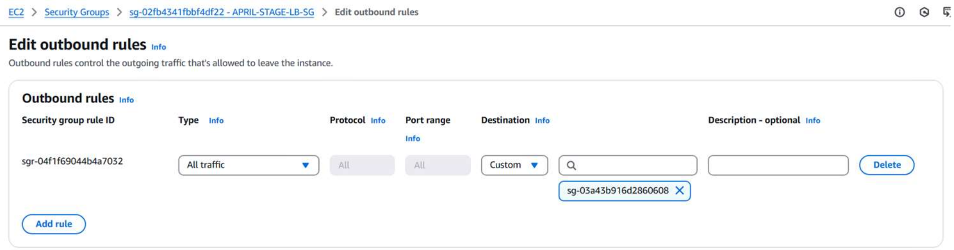
- Name: `APRIL-STAGE-ALB`
- Scheme: Internal
- IP Address Type: IPv4

➤ *Network Mapping*

- Select the same VPC as the EC2 and resources used in this Moodle Project
- For subnets, adjust as needed

➤ *Security Group*

- Create a new security group named APRIL-STAGE-LB-SG with the following rules:



EC2 > Security Groups > sg-02fb4341fbbf4df22 - APRIL-STAGE-LB-SG > Edit outbound rules

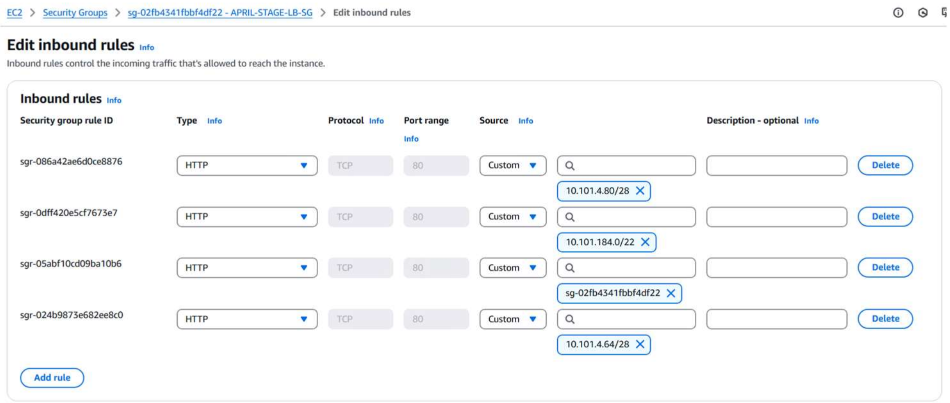
Edit outbound rules Info

Outbound rules control the outgoing traffic that's allowed to leave the instance.

Outbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Destination Info		Description - optional Info	
sgr-04f1f69044b4a7032	All traffic	All	All	Custom	sg-03a43b916d2860608		Delete

Add rule



EC2 > Security Groups > sg-02fb4341fbbf4df22 - APRIL-STAGE-LB-SG > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info		Description - optional Info	
sgr-086a42ae6d0ce8876	HTTP	TCP	80	Custom	10.101.4.80/28		Delete
sgr-0dff420e5cf7673e7	HTTP	TCP	80	Custom	10.101.184.0/22		Delete
sgr-05abf10cd09ba10b6	HTTP	TCP	80	Custom	sg-02fb4341fbbf4df22		Delete
sgr-024b9873e682ee8c0	HTTP	TCP	80	Custom	10.101.4.64/28		Delete

Add rule

➤ *Listeners and Routing*

- Protocol TCP: 8080
- Target Group: Select the target group created earlier
- For Tags, use the same as on other resources

- Click "Create load balancer"

E. MOODLE INSTALLATION

The Prima Transportasi Service Indonesia Moodle Staging Site uses version 4.5. The installation steps are officially documented in the Moodle documentation, which we reference here: Installing Moodle - MoodleDocs. In this implementation, we did not have web browser access to Moodle yet, so the installation was performed via CLI by following the reference from Administration via command line - MoodleDocs.

- Change the access permission ***/var/www/html/moodle*** so that the ***NGINX*** user user has full access rights to the directory during the installation process.

```
# chown nginx.nginx /var/www/html/moodle
# chmod -R 777 /var/www/html/moodle
```

- Create a ***moodledata*** folder in the ***/var/www*** directory and give full access rights to the ***NGINX***.

```
# mkdir -p /var/www/moodledata
# chown nginx.nginx /var/www/moodledata/ -R
# chmod -R 777 /var/www/moodledata
```

- Create config.php as a configuration requirement for Moodle installation

```
# vi /var/www/html/moodle/config.php
```

- Adjust the contents of the file to match the environment configuration

```
<?php  // Moodle configuration file

unset($CFG);
global $CFG;
$CFG = new stdClass();

$CFG->dbtype    = 'pgsql';
$CFG->dblibrary = 'native';
$CFG->dbhost    = 'aws-lab-lms-db.chqy6gyc8kng.ap-southeast-
1.rds.amazonaws.com';
$CFG->dbname    = 'moodle';
$CFG->dbuser    = 'moodleuser';
$CFG->dbpass    = 'Mo0dl3Stag3';
$CFG->prefix    = 'mdl_';
$CFG->dboptions = array (
  'dbpersist' => 0,
  'dbport' => 5432,
  'dbsocket' => '',
  'dbcollation' => 'utf8mb4_unicode_ci'
);

//Redis Configuration
$CFG->session_handler_class = '\core\session\redis';
$CFG->session_redis_host = '127.0.0.1';
$CFG->session_redis_port = 6379;
$CFG->session_redis_database = 0;
$CFG->session_redis_auth = 'moodler3d!s';
$CFG->session_redis_prefix = '';
$CFG->session_redis_acquire_lock_timeout = 120;
$CFG->session_redis_acquire_lock_warn = 0;
$CFG->session_redis_lock_expire = 7200;
$CFG->session_redis_lock_retry = 100;
$CFG->session_redis_serializer_use_igbinary = true;
$CFG->session_redis_compressor = 'gzip';

#$CFG->wwwroot   = 'http://10.101.185.54';
$CFG->wwwroot   = 'https://lms-dev.fiber.biz.id';
$CFG->dataroot  = '/var/www/moodledata';
$CFG->admin     = 'admin';

$CFG->sslproxy = true;
$CFG->reverse_proxy = true;              // Add this for ALB
$CFG->getremoteaddrconf = 0;            // Add this to properly handle client
IPs

//=========================================================================
// SETTINGS FOR DEVELOPMENT SERVERS - not intended for production use!!!
//=========================================================================
//
// Force a debugging mode regardless the settings in the site administration
@error_reporting(E_ALL | E_STRICT);   // NOT FOR PRODUCTION SERVERS!
@ini_set('display_errors', '1');         // NOT FOR PRODUCTION SERVERS!
$CFG->debug = (E_ALL | E_STRICT);   // === DEBUG_DEVELOPER - NOT FOR
PRODUCTION SERVERS!
```

```
$CFG->debugdisplay = 1;               // NOT FOR PRODUCTION SERVERS!
//
// You can specify a comma separated list of user ids that that always see
// debug messages, this overrides the debug flag in $CFG->debug and $CFG-
>debugdisplay
// for these users only.
// $CFG->debugusers = '2';
//$CFG->tool_generator_users_password = 'any123' ; // this will be the
password for the users created in next command

require_once(__DIR__ . '/lib/setup.php');

// There is no php closing tag in this file,
// it is intentional because it prevents trailing whitespace problems!
```

- Run install_database.php to install Moodle according to the configuration set in config.php

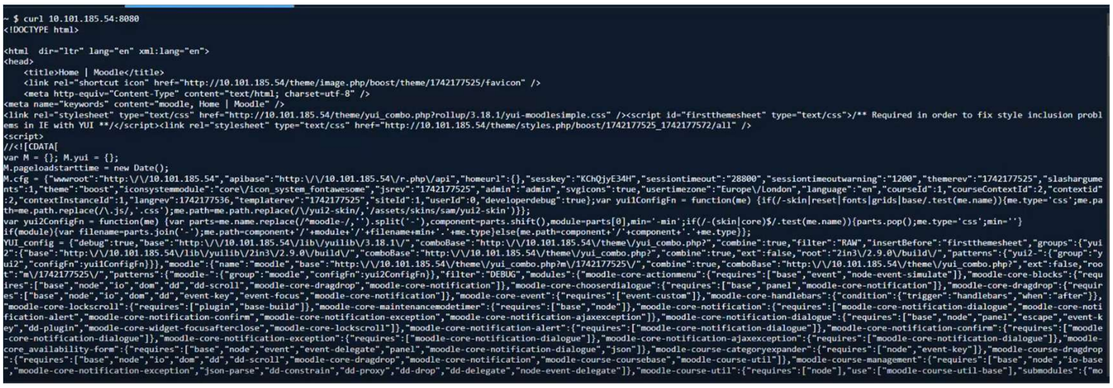
```
# php admin/cli/install_database.php --adminpass=Moodle_PTSI --
agree-license
```

- After installation is complete, restart the NGINX service

```
# systemctl restart nginx
```

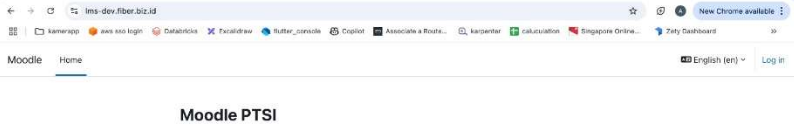
- Perform a test using curl to ensure Moodle is running

```
# curl 10.101.185.54:8080
```



```
~ $ curl 10.101.185.54:8080
<!DOCTYPE html>

<html  dir="ltr" lang="en" xml:lang="en">
<head>
    <title>Home | Moodle</title>
    <link rel="shortcut icon" href="http://10.101.185.54/theme/image.php/boost/theme/1742177525/favicon" />
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="moodle, Home | Moodle" />
<link rel="stylesheet" type="text/css" href="http://10.101.185.54/theme/yui_combo.php?rollup/3.18.1/yui-moodlesimple.css" /><script id="firstthemesheet" type="text/css">/** Required in order to fix style inclusion probl
ems in IE with YUI **/</script><link rel="stylesheet" type="text/css" href="http://10.101.185.54/theme/styles.php/boost/1742177525_1742177572/all" />
<script>
//<![CDATA[
var M = {}; M.yui = {};
M.pageloadstarttime = new Date();
M.cfg = {"wwwroot":"http:\/\/10.101.185.54","apibase":"http:\/\/10.101.185.54\/r.php\/api","homeurl":{},"sesskey":"KChQjyE34H","sessiontimeout":"28800","sessiontimeoutwarning":"1200","themerev":"1742177525","slashargume
nts":1,"theme":"boost","iconsystemmodule":"core\/icon_system_fontawesome","jsrev":"1742177525","admin":"admin","svgicons":true,"usertimezone":"Europe\/London","language":"en","courseId":1,"courseContextId":2,"contextid"
:2,"contextInstanceId":1,"langrev":1742177536,"templaterev":"1742177525","siteId":1,"userId":0,"developerdebug":true};var yui1ConfigFn = function(me) {if(/-skin|reset|fonts|grids|base/.test(me.name)){me.type='css';me.pa
th=me.path.replace(/\.js/,'.css');me.path=me.path.replace(/\/yui2-skin/,'/assets/skins/sam/yui2-skin')}};
var yui2ConfigFn = function(me) {var parts=me.name.replace(/^moodle-/,'').split('-'),component=parts.shift(),module=parts[0],min='-min';if(/-(skin|core)$/.test(me.name)){parts.pop();me.type='css';min=''}
if(module){var filename=parts.join('-');me.path=component+'/'+module+'/'+filename+min+'.'+me.type}else{me.path=component+'/'+component+'.'+me.type}};
YUI_config = {"debug":true,"base":"http:\/\/10.101.185.54\/lib\/yuilib\/3.18.1\/","comboBase":"http:\/\/10.101.185.54\/theme\/yui_combo.php?","combine":true,"filter":"RAW","insertBefore":"firstthemesheet","groups":{"yui
2":{"base":"http:\/\/10.101.185.54\/lib\/yuilib\/2in3\/2.9.0\/build\/","comboBase":"http:\/\/10.101.185.54\/theme\/yui_combo.php?","combine":true,"ext":false,"root":"2in3\/2.9.0\/build\/","patterns":{"yui2-":{"group":"y
ui2","configFn":yui1ConfigFn}}},"moodle":{"name":"moodle","base":"http:\/\/10.101.185.54\/theme\/yui_combo.php?m\/1742177525\/","combine":true,"comboBase":"http:\/\/10.101.185.54\/theme\/yui_combo.php?","ext":false,"roo
t":"m\/1742177525\/","patterns":{"moodle-":{"group":"moodle","configFn":yui2ConfigFn}},"filter":"DEBUG","modules":{"moodle-core-actionmenu":{"requires":["base","event","node-event-simulate"]},"moodle-core-blocks":{"requ
ires":["base","node","io","dom","dd","dd-scroll","moodle-core-dragdrop","moodle-core-notification"]},"moodle-core-chooserdialogue":{"requires":["base","panel","moodle-core-notification"]},"moodle-core-dragdrop":{"requir
es":["base","node","io","dom","dd","event-key","event-focus","moodle-core-notification"]},"moodle-core-event":{"requires":["event-custom"]},"moodle-core-handlebars":{"condition":{"trigger":"handlebars","when":"after"}},
"moodle-core-lockscroll":{"requires":["plugin","base-build"]},"moodle-core-maintenancemodetimer":{"requires":["base","node"]},"moodle-core-notification":{"requires":["moodle-core-notification-dialogue","moodle-core-noti
fication-alert","moodle-core-notification-confirm","moodle-core-notification-exception","moodle-core-notification-ajaxexception"]},"moodle-core-notification-dialogue":{"requires":["base","node","panel","escape","event-k
ey","dd-plugin","moodle-core-widget-focusafterclose","moodle-core-lockscroll"]},"moodle-core-notification-alert":{"requires":["moodle-core-notification-dialogue"]},"moodle-core-notification-confirm":{"requires":["moodle
-core-notification-dialogue"]},"moodle-core-notification-exception":{"requires":["moodle-core-notification-dialogue"]},"moodle-core-notification-ajaxexception":{"requires":["moodle-core-notification-dialogue"]},"moodle-
core_availability-form":{"requires":["base","node","event","event-delegate","panel","moodle-core-notification-dialogue","json"]},"moodle-course-categoryexpander":{"requires":["node","event-key"]},"moodle-course-dragdrop
":{"requires":["base","node","io","dom","dd","dd-scroll","moodle-core-dragdrop","moodle-core-notification","moodle-course-coursebase","moodle-course-util"]},"moodle-course-management":{"requires":["base","node","io-base
","moodle-core-notification-exception","json-parse","dd-constrain","dd-proxy","dd-drop","dd-delegate","node-event-delegate"]},"moodle-course-util":{"requires":["node"],"use":["moodle-course-util-base"],"submodules":{"mo
```

- Open through a web browser after getting the Moodle application exposed to the public.

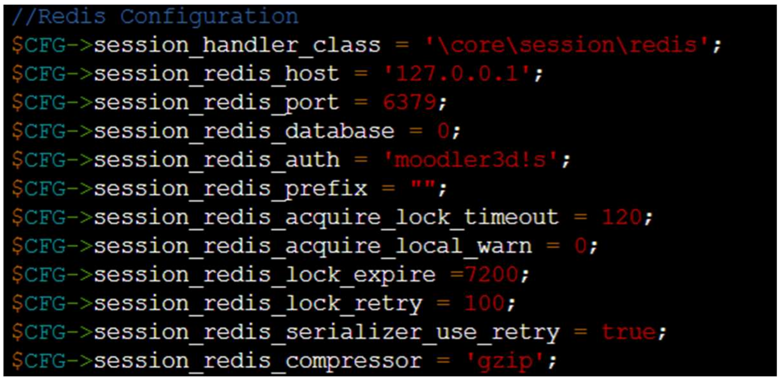


F. POST-INSTALLATION CONFIGURATION

After installing Moodle, there are several steps that need to be taken to improve Moodle performance, including securing the Moodle site (https), hardening NGINX, tuning MariaDB, and tuning PHP-FPM.

1. Redis Configuration in Moodle

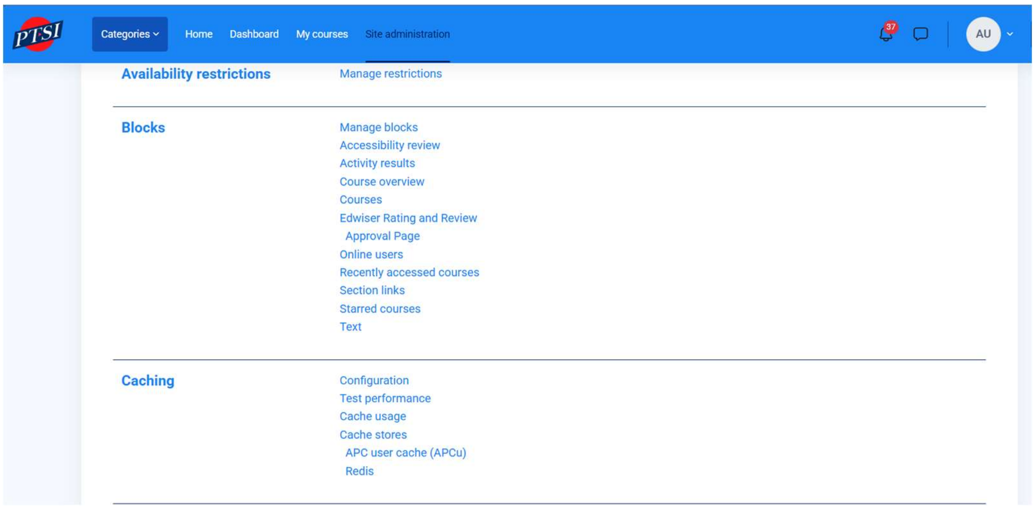
- Ensure Redis configuration is set in config.php



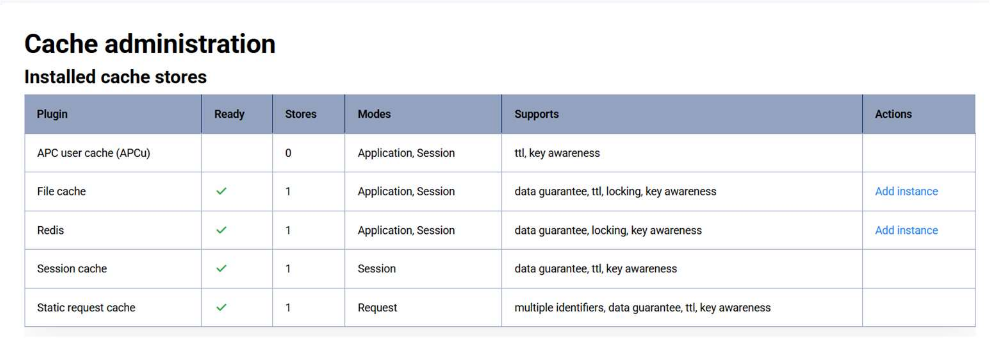
```
//Redis Configuration
$CFG->session_handler_class = '\core\session\redis';
$CFG->session_redis_host = '127.0.0.1';
$CFG->session_redis_port = 6379;
$CFG->session_redis_database = 0;
$CFG->session_redis_auth = 'moodler3d!s';
$CFG->session_redis_prefix = "";
$CFG->session_redis_acquire_lock_timeout = 120;
$CFG->session_redis_acquire_local_warn = 0;
$CFG->session_redis_lock_expire =7200;
$CFG->session_redis_lock_retry = 100;
$CFG->session_redis_serializer_use_retry = true;
$CFG->session_redis_compressor = 'gzip';
```

- Log in to Moodle through a web browser using administrator credentials

- Open Site administration > Caching > Configuration



- Make sure Redis has a check mark in the Ready column. If so, in the Actions column, select “Add instance”.

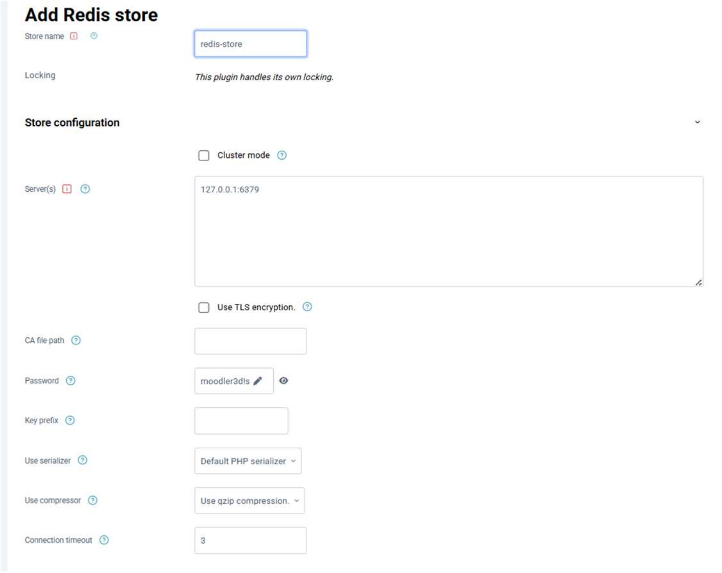


Cache administration

Installed cache stores

Plugin	Ready	Stores	Modes	Supports	Actions
APC user cache (APCu)		0	Application, Session	ttl, key awareness	
File cache	✓	1	Application, Session	data guarantee, ttl, locking, key awareness	Add instance
Redis	✓	1	Application, Session	data guarantee, locking, key awareness	Add instance
Session cache	✓	1	Session	data guarantee, ttl, key awareness	
Static request cache	✓	1	Request	multiple identifiers, data guarantee, ttl, key awareness	

- Fill in the Redis configuration according to config.php redis configuration sesuai dengan config.php



Add Redis store

Store name	redis-store
Locking	*This plugin handles its own locking.*

Store configuration

	□ Cluster mode
Server(s)	127.0.0.1:6379
	□ Use TLS encryption.
CA file path	
Password	moodler3d!s
Key prefix	
Use serializer	Default PHP serializer
Use compressor	Use gzip compression.
Connection timeout	3

2. Redis Tuning Configuration

- Open the ***/etc/redis/redis.conf*** file.

```
# vi /etc/redis/redis.conf
```

- Find and change the configuration lines to the following. Remove the hash symbol on the following configuration lines.

```
bind 127.0.0.1 -::1
supervised systemd
requirepass outputgeneratepassword
```

- Adjust the following lines in the configuration file.

```
maxclients 10000
maxmemory 4gb
maxmemory-policy allkeys-lru
```

- Restart the Redis service after making these configurations and ensure the Redis service is running.

```
# systemctl restart redis
```

- Make sure the Redis service is listening on localhost (127.0.0.1) according to the configuration performed.

```
# netstat -tulpn | grep redis
```

- Purge all caches in Moodle.

```
# php /var/www/html/moodle/admin/cli/purge_caches.php
```

3. NGINX Tuning and Hardening Configuration

- This configuration is done to enhance the security of the NGINX service for Moodle. The security aspects include adding FastCGI, Routing Engine, Hiding internal files, and XSendfile.

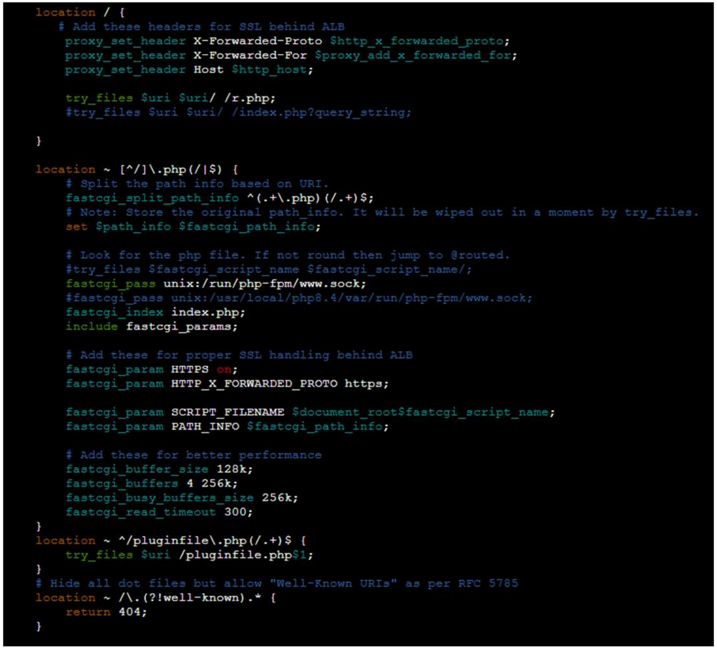
- Open the NGINX configuration file for the Moodle app at ***/etc/nginx/conf.d/moodle.conf***.

```
# vi /etc/nginx/conf.d/moodle.conf
```

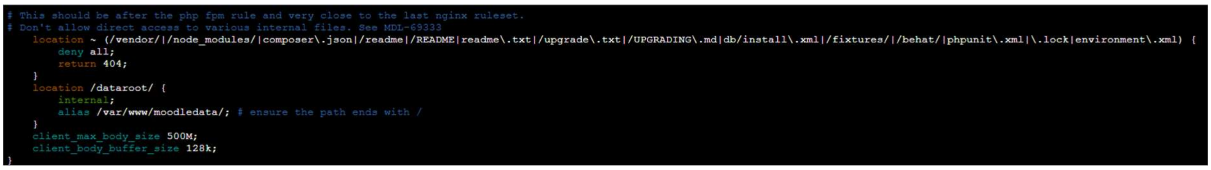
- Add the following configurations according to documentation recommendations

```
#FastCGI Configuration
location ~ \.php(/|$) {
  # Split the path info based on URI.
  fastcgi_split_path_info ^(.+\.php)(/.*)$;

  # Look for the php file. If not round then jump to @routed.
  try_files $fastcgi_script_name $fastcgi_script_name/;

  # File was found - pass to fastcgi.
  fastcgi_pass   127.0.0.1:9000;
  include        fastcgi_params;

  # Re-apply the path_info after including fastcgi_params.
  fastcgi_param PATH_INFO $path_info;
  fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;
  fastcgi_param DOCUMENT_ROOT $realpath_root;
}

#Routing Engine
location / {
    try_files $uri /r.php;
}
# Hide all dot files but allow "Well-Known URIs" as per RFC 5785
location ~ /\.(?!well-known).* {
    return 404;
}

# This should be after the php fpm rule and very close to the last
nginx ruleset.
# Don't allow direct access to various internal files. See MDL-
69333
location ~
(/vendor/|/node_modules/|composer\.json|/readme|/README|readme\.txt
|/upgrade\.txt|/UPGRADING\.md|db/install\.xml|/fixtures/|/behat/|ph
punit\.xml|\.lock|environment\.xml) {
    deny all;
    return 404;
}

# XSendfile aka X-Accel-Redirect
location /dataroot/ {
    internal;
    alias <full_moodledata_path>; # ensure the path ends with /
}
```

- Here is an example implementation of these configurations in NGINX



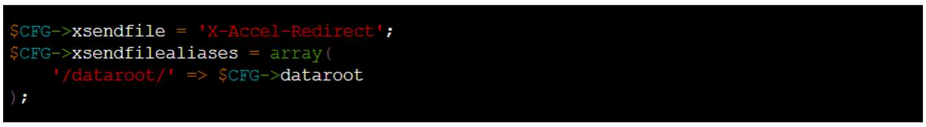
```
location / {
   # Add these headers for SSL behind ALB
    proxy_set_header X-Forwarded-Proto $http_x_forwarded_proto;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;

    try_files $uri $uri/ /r.php;
    #try_files $uri $uri/ /index.php?query_string;

}

location ~ [^/]\.php(/|$) {
    # Split the path info based on URI.
    fastcgi_split_path_info ^(.+\.php)(/.+)$;
    # Note: Store the original path_info. It will be wiped out in a moment by try_files.
    set $path_info $fastcgi_path_info;

    # Look for the php file. If not round then jump to @routed.
    #try_files $fastcgi_script_name $fastcgi_script_name/;
    fastcgi_pass unix:/run/php-fpm/www.sock;
    #fastcgi_pass unix:/usr/local/php8.4/var/run/php-fpm/www.sock;
    fastcgi_index index.php;
    include fastcgi_params;

    # Add these for proper SSL handling behind ALB
    fastcgi_param HTTPS on;
    fastcgi_param HTTP_X_FORWARDED_PROTO https;

    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    fastcgi_param PATH_INFO $fastcgi_path_info;

    # Add these for better performance
    fastcgi_buffer_size 128k;
    fastcgi_buffers 4 256k;
    fastcgi_busy_buffers_size 256k;
    fastcgi_read_timeout 300;
}
location ~ ^/pluginfile\.php(/.+)$ {
    try_files $uri /pluginfile.php$1;
}
# Hide all dot files but allow "Well-Known URIs" as per RFC 5785
location ~ /\.(?!well-known).* {
    return 404;
}
```



```
# This should be after the php fpm rule and very close to the last nginx ruleset.
# Don't allow direct access to various internal files. See MDL-69333
   location ~ (/vendor/|/node_modules/|composer\.json|/readme|/README|readme\.txt|/upgrade\.txt|/UPGRADING\.md|db/install\.xml|/fixtures/|/behat/|phpunit\.xml|\.lock|environment\.xml) {
       deny all;
       return 404;
   }
   location /dataroot/ {
       internal;
       alias /var/www/moodledata/; # ensure the path ends with /
   }
   client_max_body_size 500M;
   client_body_buffer_size 128k;
}
```

- Enable XSendfile config.php

```
# vi /var/www/html/moodle/config.php
```



```
$CFG->xsendfile = 'X-Accel-Redirect';
$CFG->xsendfilealiases = array(
    '/dataroot/' => $CFG->dataroot
);
```

- Restart the NGINX service

```
# systemctl restart nginx
```

4. PHP-FPM Tuning Configuration

- Open the ***/etc /php-fpm.d/www.conf*** file

```
# vi /etc/php-fpm.d/www.conf
```

- Find and change the configuration lines to the following. Make sure the configuration lines are not commented out.

```
pm = static
pm.max_children = 200
pm.max_requests = 10000
```

- Close and save the file then open the php.ini file

```
# vi /etc/php.ini
```

- Find and change the configuration lines to the following:

```
post_max_size = 1024M
upload_max_filesize = 1024M
memory_limit = 512M
max_file_upload = 100
```

- Restart the php-fpm and NGINX services.

```
# systemctl restart nginx && systemctl restart php-fpm
```

5. Updating Moodle Directory Permissions

- Change permissions on the ***/var/www/html/moodle.***

```
# chown -R root:nginx /var/www/html/moodle
# chmod -R 0755 /var/www/html/moodle
```

6. Mounting S3 Bucket

- Mount the S3 Bucket as a filesystem on the Moodle server so that users can upload SCORM Packages directly to S3. Moodle can then access and read these files through the configured mount point aspart of the local filesystem.

- Execute the following command to mount the S3 bucket:

```
# mount-s3 aws-stage-lms-data-s3 /var/www/moodledata/repository --
uid=992 --gid=992 --allow-other
```